WHAT IS CLAIMED IS:

1. A floating point max/min circuit for determining a threshold condition between two floating point operands, comprising:

a first analysis circuit configured to determine a format of a first floating point operand of the two floating point operands based upon floating point status information encoded within the first floating point operand;

a second analysis circuit configured to determine a format of a second floating point operand of the two floating point operands based upon floating point status information encoded within the second floating point operand;

a decision circuit, coupled to the first analysis circuit and to the second analysis circuit and responding to a function control signal that indicates the threshold condition is one of a maximum of the two floating point operands and a minimum of the two floating point operands, for generating at least one assembly control signal based on the format of a first floating point operand, the format of a second floating point operand, and the function control signal; and

a result assembler circuit, coupled to the decision circuit, for producing a result indicating which of the first floating point operand and the second floating point operand meet the threshold condition, based on the at least one assembly control signal.

2. The floating point max/min circuit of claim 1 further comprising:

a first operand buffer coupled to the first analysis circuit, the first operand buffer supplying the first floating point operand to the first analysis circuit; and

a second operand buffer coupled to the second analysis circuit, the second operand buffer supplying the second floating point operand to the second analysis circuit.

3. The floating point max/min circuit of claim 1, wherein the format is from a group comprising: not-a-number (NaN), positive infinity, negative infinity, normalized, denormalized, positive overflow, negative overflow, positive underflow, negative underflow, inexact, exact, division by zero, invalid operation, positive zero, and negative zero.

wherein the normalized and denormalized formats indicate a finite numerical value.

4. The floating point max/min circuit of claim 3, wherein the decision circuit treats the first floating point operand as less than the second floating point operand if the format of the first floating point indicates positive overflow and the format of the second floating point operand indicates positive infinity, and

wherein the decision circuit treats the first floating point operand as greater than the second floating point operand if the format of the first floating point indicates positive overflow and the format of the second floating point operand indicates a finite numerical value.

5. The floating point max/min circuit of claim 3, wherein the decision circuit treats the first floating point operand as greater than the second floating point operand if the format of the first floating point indicates negative overflow and the format of the second floating point operand indicates negative infinity, and

wherein the decision circuit treats the first floating point operand as less than the second floating point operand if the format of the first floating point indicates negative overflow and the format of the second floating point operand indicates a finite numerical value.

6. The floating point max/min circuit of claim 3, wherein the decision circuit treats the first floating point operand as greater than a second floating point operand if the format of the first floating point indicates positive underflow and the format of the second floating point operand indicates positive zero, and

wherein the decision circuit treats the first floating point operand as less than a second floating point operand if the format of the first floating point indicates positive underflow and the format of the second floating point operand indicates a positive finite numerical value.

7. The floating point max/min circuit of claim 3, wherein the decision circuit treats the first floating point operand as less than the second floating point operand if the format of the first floating point indicates negative overflow and the format of the second floating point operand indicates negative zero, and

wherein the decision circuit treats the first floating point operand as greater than the second floating point operand if the format of the first floating point indicates negative underflow and the format of the second floating point operand indicates a negative finite numerical value.

8. The floating point max/min circuit of claim 3, wherein the floating point max/min circuit obeys a commutative law of arithmetic when the format of at least one of the two floating point operands indicates not-a-number (NaN).

9. The floating point max/min circuit of claim 3, wherein the floating point max/min circuit obeys an associative law of arithmetic when the format of at least one of the two floating point operands indicates not-a-number (NaN).

10. The floating point max/min circuit of claim 3, wherein if the status of at least one of the two floating point operands indicates not-a-number (NaN), then the decision circuit obeys the identities:

$$\max(-x,-y) = -\min(x,y)$$

and $$\min(-x,-y) = -\max(x,y);$$

wherein x represents a value of first floating point operand and y represents a value of the second floating point operand, and

wherein the negation operation "-x" complements the sign bit of x.

11. The floating point max/min circuit of claim 3, wherein if the format of at least one of the floating point operands indicates not-a-number (NaN), then the decision circuit obeys the identities:

$$\max(\min(x,y), \min(x,z)) = \min(x,\max(y,z)), \text{ and}$$

$$\min(\max(x,y), \max(x,z)) = \max(x,\min(y, z));$$

wherein x represents a value of the first floating point operand, y represents a first value of the second floating point operand, and z represents a second value of the second floating point operand.

12. The floating point max/min circuit of claim 3, wherein if the respective formats of the two floating point operands indicate not-a-number (NaN), then the decision circuit dynamically determines which of the two NaN floating point operands to represent in the result.

13. The floating point max/min circuit of claim 3, wherein if the respective formats of the two floating point operands indicate not-a-number (NaN), then the floating-point max/min circuit produces the result using the floating point status information from the one of the two floating point operands having a larger fraction.

14. The floating point max/min circuit of claim 3, wherein the result is a third floating point operand having encoded floating point status information.

15. The floating point max/min circuit of claim 14, wherein at least part of the encoded floating point status information in the result comes from at least one of: the first floating point operand and the second floating point operand.

16. The floating point max/min circuit of claim 14, wherein the encoded floating point status information in the result further comprises overflow status information if the encoded floating point status information of one of the two floating point operands indicates overflow status and the encoded floating point status information of the other of the two floating point operands indicates at least one of: an infinity status and a NaN status.

17. The floating point max/min circuit of claim 14, wherein the encoded floating point status information in the result further comprises underflow status information if the encoded floating point status information of one of the two floating point operands indicates underflow status and the encoded floating point status information of the other of the two floating point operands indicates at least one of: an infinity status and a NaN status.

18. The floating point max/min circuit of claim 14, wherein the encoded floating point status information in the result further comprises inexact status information if the encoded floating point status information of one of the two floating point operands indicates overflow status and the encoded floating point status information of the other of the two floating point operands indicates at least one of: an infinity status and a NaN status.

19. The floating point max/min circuit of claim 14, wherein the encoded floating point status information in the result further comprises inexact status information if the encoded floating point status information of one of the two floating point operands indicates underflow status and the encoded floating point status information of the other of the two floating point operands indicates at least one of: an infinity status and a NaN status.

20. The floating point max/min circuit of claim 14, wherein if the format of the first floating point operand indicates NaN and the format of the second floating point operand indicates infinity, then the result produced is in the NaN format and contains floating point status information that is a combination of the floating point status information encoded within the first

floating point operand and the floating point status information encoded within the second floating point operand.

21. The floating point max/min circuit of claim 14, wherein if the format of the first floating point operand indicates NaN and the format of the second floating point operand indicates overflow, then the result produced is in the NaN format and contains floating point status information that is a combination of the floating point status information encoded within the first floating point operand and overflow status information.

22. The floating point max/min circuit of claim 14, wherein if the format of the first floating point operand indicates NaN and the format of the second floating point operand indicates underflow, then the result produced is in the NaN format and contains floating point status information that is a combination of the floating point status information encoded within the first floating point operand and underflow status information.

23. The floating point max/min circuit of claim 14, wherein if the format of the first floating point operand indicates NaN and the format of the second floating point operand indicates overflow, then the result produced is in the NaN format and contains floating point status information that is a combination of the floating point status information encoded within the first floating point operand and inexact status information.

24. The floating point max/min circuit of claim 14, wherein if the format of the first floating point operand indicates NaN and the format of the second floating point operand

indicates underflow, then the result produced is in the NaN format and contains floating point status information that is a combination of the floating point status information encoded within the first floating point operand and inexact status information.

25. The floating point max/min circuit of claim 14, wherein if the format of the first floating point operand indicates NaN and the format of the second floating point operand indicates NaN, then the result produced is in the NaN format and contains floating point status information that is a combination of the floating point status information encoded within the first floating point operand and the floating point status information encoded within the second floating point operand.

26. The floating point max/min circuit of claim 14, wherein if the format of the first floating point operand indicates infinity and the format of the second floating point operand indicates infinity, then the result produced contains floating point status information that is a combination of the floating point status information encoded within the first floating point operand and the floating point status information encoded within the second floating point operand.

27. The floating point max/min circuit of claim 14, wherein if the format of the first floating point operand indicates infinity and the format of the second floating point operand indicates overflow, then the result produced is a copy of the first floating point operand in the infinity format.

28. The floating point max/min circuit of claim 14, wherein if the format of the first floating point operand indicates infinity and the format of the second floating point operand indicates overflow, then the result produced is a copy of the first floating point operand in the infinity format.

29. A method for providing one of the maximum and the minimum of a first floating point operand and a second floating point operand, comprising:

receiving the first floating point operand;

receiving the second floating point operand;

determining a format of the first floating point operand based upon floating point status information encoded within the first floating point operand;

determining a format of the second floating point operand based upon floating point status information encoded within the second floating point operand; and

producing a result indicating which of the floating point operand and the second floating point operand is larger or smaller than the other, based on the first format and the second format.

30. The method of claim 29 further comprising:

receiving at least one control signal that causes the result to indicate one of: the maximum of the first and second floating point operands and the minimum of the first and second floating point operands.

31. The method of claim 29, wherein the format determined for the first floating point operand and the format determined for the second floating point are from a group comprising:

not-a-number (NaN), positive infinity, negative infinity, normalized, denormalized, positive overflow, negative overflow, positive underflow, negative underflow, inexact, exact, division by zero, invalid operation, positive zero, and negative zero.

32. The method of claim 29, wherein the format represents one of a positive overflow (+OV) and a negative overflow (-OV).

33. The method of claim 29, wherein the format represents one of a positive underflow (+UN) and a negative underflow (-UN).

34. The method of claim 29, wherein the format represents one of a positive infinity and a negative infinity.

35. The method of claim 29, wherein the step of producing a result treats the first floating point operand as less than the second floating point operand if the format of the first floating point indicates positive overflow and the format of the second floating point operand indicates positive infinity, and

wherein the step of producing a result treats the first floating point operand as greater than the second floating point operand if the format of the first floating point indicates positive overflow and the format of the second floating point operand indicates a finite numerical value.

36. The method of claim 29, wherein the step of producing a result treats the first floating point operand as greater than the second floating point operand if the format of the first

floating point indicates negative overflow and the format of the second floating point operand indicates negative infinity, and

wherein the step of producing a result treats the first floating point operand as less than the second floating point operand if the format of the first floating point indicates negative overflow and the format of the second floating point operand indicates a finite numerical value.

37. The method of claim 29, wherein the step of producing a result treats the first floating point operand as greater than a second floating point operand if the format of the first floating point indicates positive underflow and the format of the second floating point operand indicates positive zero, and

wherein the step of producing a result treats the first floating point operand as less than a second floating point operand if the format of the first floating point indicates positive underflow and the format of the second floating point operand indicates a positive finite numerical value.

38. The method of claim 29, wherein the step of producing a result treats the first floating point operand as less than the second floating point operand if the format of the first floating point indicates negative overflow and the format of the second floating point operand indicates negative zero, and

wherein the step of producing a result treats the first floating point operand as greater than the second floating point operand if the format of the first floating point indicates negative underflow and the format of the second floating point operand indicates a negative finite numerical value.

39. The method of claim 29, wherein the step of producing a result further comprises:

obeying a commutative law of arithmetic when the format of at least one of the two floating point operands indicates not-a-number (NaN).

40. The method of claim 29, wherein the step of producing a result further comprises:

obeying an associative law of arithmetic if the format of at least one of the two floating point operands indicates not-a-number (NaN).

41. The method of claim 29, wherein if the status of at least one of the two floating point operands indicates not-a-number (NaN), then the step of producing a result obeys the identities:

$$\max(-x,-y) = -\min(x,y)$$

and $\min(-x,-y) = -\max(x,y)$;

wherein x represents the first floating point operand and y represents the second floating point operand, and

wherein the negation operation "-x" complements the sign bit of x.

42. The method of claim 29, wherein if the format of at least one of the floating point operands indicates not-a-number (NaN), then the step of producing a result obeys the identities:

$$\max(\min(x,y), \min(x,z)) = \min(x,\max(y,z)), \text{ and}$$

$$\min(\max(x,y), \max(x,z)) = \max(x,\min(y,z));$$

wherein x represents the first floating point operand, y represents the second floating point operand, and z represents a third floating point operand.

43. The method of claim 29, wherein if the respective formats of the two floating point operands indicate not-a-number (NaN), then the step of producing the result further comprises the step of:

using the floating point status information from the one of the two floating point operands having a larger fraction to produce the result.

44. The method of claim 29, wherein the result is a third floating point operand having encoded floating point status information.

45. The method of claim 44, wherein if the format of the first floating point operand indicates NaN and the format of the second floating point operand indicates infinity, then the result produced is in the NaN format and contains floating point status information that is a combination of the floating point status information encoded within the first floating point operand and the floating point status information encoded within the second floating point operand.

46. The method of claim 44, wherein if the format of the first floating point operand indicates NaN and the format of the second floating point operand indicates overflow, then the result produced is in the NaN format and contains floating point status information that is a combination of the floating point status information encoded within the first floating point operand and overflow status information.

47. The method of claim 44, wherein if the format of the first floating point operand indicates NaN and the format of the second floating point operand indicates underflow, then the result produced is in the NaN format and contains floating point status information that is a combination of the floating point status information encoded within the first floating point operand and underflow status information.

48. The method of claim 44, wherein if the format of the first floating point operand indicates NaN and the format of the second floating point operand indicates overflow, then the result produced is in the NaN format and contains floating point status information that is a combination of the floating point status information encoded within the first floating point operand and inexact status information.

49. The method of claim 44, wherein if the format of the first floating point operand indicates NaN and the format of the second floating point operand indicates underflow, then the result produced is in the NaN format and contains floating point status information that is a combination of the floating point status information encoded within the first floating point operand and inexact status information.

50. The method of claim 44, wherein if the format of the first floating point operand indicates NaN and the format of the second floating point operand indicates NaN, then the result produced is in the NaN format and contains floating point status information that is a combination of the floating point status information encoded within the first floating point

operand and the floating point status information encoded within the second floating point operand.

51. The method of claim 44, wherein if the format of the first floating point operand indicates infinity and the format of the second floating point operand indicates infinity, then the result produced contains floating point status information that is a combination of the floating point status information encoded within the first floating point operand and the floating point status information encoded within the second floating point operand.

52. The method of claim 44, wherein if the format of the first floating point operand indicates infinity and the format of the second floating point operand indicates overflow, then the result produced is a copy of the first floating point operand in the infinity format.

53. The method of claim 44, wherein if the format of the first floating point operand indicates infinity and the format of the second floating point operand indicates overflow, then the result produced is a copy of the first floating point operand in the infinity format.

54. The method of claim 44, wherein at least part of the encoded floating point status information in the result comes from at least one of: the first floating point operand and the second floating point operand.

55. The method of claim 54, wherein the encoded floating point status information in the result further comprises overflow status information if the encoded floating point status

information of one of the two floating point operands indicates overflow status and the encoded floating point status information of the other of the two floating point operands indicates at least one of: an infinity status and a NaN status.

56. The method of claim 54, wherein the encoded floating point status information in the result further comprises underflow status information if the encoded floating point status information of one of the two floating point operands indicates underflow status and the encoded floating point status information of the other of the two floating point operands indicates at least one of: an infinity status and a NaN status.

57. The method of claim 54, wherein the encoded floating point status information in the result further comprises inexact status information if the encoded floating point status information of one of the two floating point operands indicates overflow status and the encoded floating point status information of the other of the two floating point operands indicates at least one of: an infinity status and a NaN status.

58. The method of claim 54, wherein the encoded floating point status information in the result further comprises inexact status information if the encoded floating point status information of one of the two floating point operands indicates underflow status and the encoded floating point status information of the other of the two floating point operands indicates at least one of: an infinity status and a NaN status.

59. A computer-readable medium on which is stored a set of instructions for providing the maximum or minimum of first floating point operand and a second floating point operand, which when executed perform steps comprising:

receiving the first floating point operand;

receiving the second floating point operand;

determining a format of the first floating point operand based upon floating point status information encoded within the first floating point operand;

determining a format of the second floating point operand based upon floating point status information encoded within the second floating point operand; and

producing a result indicating which of the floating point operand and the second floating point operand is larger or smaller than the other, based on the first format and the second format.

60. The computer-readable medium of claim 59, wherein the format determined for the first floating point operand and the format determined for the second floating point are from a group comprising: not-a-number (NaN), positive infinity, negative infinity, normalized, denormalized, positive overflow, negative overflow, positive underflow, negative underflow, inexact, exact, division by zero, invalid operation, positive zero, and negative zero.

61. The computer-readable medium of claim 59, wherein the result is a third floating point operand having encoded floating point status information.

62. The computer-readable medium of claim 59, wherein at least part of the encoded floating point status information in the result comes from at least one of: the first floating point operand and the second floating point operand.

63. The computer-readable medium of claim 59, wherein the step of producing a result treats the first floating point operand as less than the second floating point operand if the format of the first floating point indicates positive overflow and the format of the second floating point operand indicates positive infinity, and

wherein the step of producing a result treats the first floating point operand as greater than the second floating point operand if the format of the first floating point indicates positive overflow and the format of the second floating point operand indicates a finite numerical value.

64. The computer-readable medium of claim 59, wherein the step of producing a result treats the first floating point operand as greater than the second floating point operand if the format of the first floating point indicates negative overflow and the format of the second floating point operand indicates negative infinity, and

wherein the step of producing a result treats the first floating point operand as less than the second floating point operand if the format of the first floating point indicates negative overflow and the format of the second floating point operand indicates a finite numerical value.

65. The computer-readable medium of claim 59, wherein the step of producing a result treats the first floating point operand as greater than a second floating point operand if the

format of the first floating point indicates positive underflow and the format of the second floating point operand indicates positive zero, and

wherein the step of producing a result treats the first floating point operand as less than a second floating point operand if the format of the first floating point indicates positive underflow and the format of the second floating point operand indicates a positive finite numerical value.

66. The computer-readable medium of claim 59, wherein the step of producing a result treats the first floating point operand as less than the second floating point operand if the format of the first floating point indicates negative overflow and the format of the second floating point operand indicates negative zero, and

wherein the step of producing a result treats the first floating point operand as greater than the second floating point operand if the format of the first floating point indicates negative underflow and the format of the second floating point operand indicates a negative finite numerical value.

67. The computer-readable medium of claim 59, wherein the step of producing a result further comprises:

obeying a commutative law of arithmetic when the format of at least one of the two floating point operands indicates not-a-number (NaN).

68. The computer-readable medium of claim 59, wherein the step of producing a result further comprises:

obeying an associative law of arithmetic if the format of at least one of the two floating point operands indicates not-a-number (NaN).

69. The computer-readable medium of claim 59, wherein if the status of at least one of the two floating point operands indicates not-a-number (NaN), then the step of producing a result obeys the identities:

$$\max(-x,-y) = -\min(x,y)$$

and $$\min(-x,-y) = -\max(x,y);$$

wherein x represents the first floating point operand and y represents the second floating point operand, and

wherein the negation operation "-x" complements the sign bit of x.

70. The computer-readable medium of claim 59, wherein if the format of at least one of the floating point operands indicates not-a-number (NaN), then the step of producing a result obeys the identities:

$$\max(\min(x,y), \min(x,z)) = \min(x,\max(y,z)), \text{ and}$$

$$\min(\max(x,y), \max(x,z)) = \max(x,\min(y, z));$$

wherein x represents the first floating point operand, y represents the second floating point operand, and z represents a third floating point operand.

71. The computer-readable medium of claim 59, wherein if the respective formats of the two floating point operands indicate not-a-number (NaN), then the step of producing the result further comprises the step of:

using the floating point status information from the one of the two floating point operands having a larger fraction to produce the result.